PMappingTab Instruction Manual

David Corsar, Derek Sleeman

First Draft

July 8, 2005

Contents

1 Introduction **2**

2 Setting up PJMappingTab **2**
2.1 Installing PJMappingTab 2
2.2 Including PJMappingTab in the Current Protégé Project 2
2.3 Removing PJMappingTab from the Current Protégé Project . . . 3

3 Using PJMappingTab **3**
3.1 Performing Rule Abstraction 3
3.2 Importing The Ontology 3
3.2.1 Importing The Current Ontology 4
3.2.2 Importing The Ontology from Another Project 4
3.3 Importing Rules 4
3.4 Creating Abstract Rules 4
3.5 Saving Abstract Rules 5
3.6 Performing Rule Mapping 5
3.7 Importing The Ontology to Map To 5
3.8 Importing Abstract Rules 6
3.9 Mapping 6
3.9.1 Automatic Mapping 6
3.9.2 Setting the Automatic Mapping Properties 7
3.9.3 Manual Mapping 7
3.10 Executing the Mappings 9
3.10.1 Saving the New Rules 9

1 Introduction

PJMappingTab (Protégé-Jess Mapping Tab) provides an environment which enables the reuse of JessTab rules across multiple ontologies from within the Protégé-2.1 ontology creation and editor environment. This document provides instructions on how to use the various facilities of PJMappingTab to achieve this.

2 Setting up PJMappingTab

The first section of this manual covers installing PJMappingTab, how to include it in Protégé project and then how to remove it.

2.1 Installing PJMappingTab

Before PJMappingTab can be used, the relevant files need to be installed into the Protégé plugins folder. Assuming starting with the pjmappingtab.zip file

1. Unzip the contents of **pjmappingtab.zip** into the **plugins** folder of the Protégé directory (for Protégé-2.1 this is typically located somewhere like **C:\Program Files\Protege-2.1\plugins**),
2. The plugins folder should now contain a folder called **uk.ac.csd.abdn.pjmappingtab** which should contain files named "jazzy-core.jar", "JWord.jar" and "pjmappingtab.jar" and a folder named "wordnetdata", which should contain the 21 WordNet dictionary files.

2.2 Including PJMappingTab in the Current Protégé Project

Assuming that PJMappingTab and its associated files have been installed correctly, including PJMappingTab in the current Protégé project is achieved in the same manner as adding any plug-in:

1. From the Protégé menu bar select **Project**,
2. Select **Configure**,
3. In the following dialog box, ensure the **Tab Widgets** tab is select. This tab displays all the tab plug-ins that can be included in the project. Click the check box next to **PJMappingTab** - a tick (or cross) should appear in the box to signal it has been selected,
4. Press **OK**.

PJMappingTab should be now be displayed beside the other tabs in the main Protégé window.

2.3 Removing PJMappingTab from the Current Protégé Project

PJMappingTab can be removed from the Protégé project at any time, just like any other tab. To achieve this:

1. Select **Project**,
2. Select **Configure**,
3. In the following dialog box, ensure the **Tab Widgets** tab is select. This tab displays all the tab plug-ins that can be included in the project. The check box next to **PJMappingTab** should have a tick in it to show it is currently included, click on it to remove it,
4. Press **OK**.

PJMappingTab has now been removed and should not feature in the Protégé window.

3 Using PJMappingTab

This section describes how PJMappingTab can be used to enable the reuse of a set of JessTab rules against different ontologies within the Protégé environment. Once a set of JessTab rules have been developed for use against one ontology, PJMappingTab can help prepare the same set of rules for use against another ontology. This process comprises of two phases - the rule abstraction phase and the rule mapping phase. How these phases are achieved with PJMappingTab is described below.

3.1 Performing Rule Abstraction

The following section describes the rule abstraction phase of the process. In this phase a set of JessTab rules designed to be ran against one ontology, are combined with that ontology to make a set of abstract rules. These abstract rules are later used in the mapping phase where they are tailored so they can be run against another ontology.

3.2 Importing The Ontology

For the abstraction process to be successful PJMappingTab requires both the set of rules and the ontology that they were designed for as it uses the class and slot names in the ontology, along with the slot data types to perform the rule abstraction.

The ontology can be imported from one of two locations: the project currently loaded in Protégé (i.e. the one displayed in the **Classes** tab) or from another Protégé project saved on disk.

3.2.1 Importing The Current Ontology

To import the current ontology, in the **Original** tab, press the **Import from Current KB** button. The ontology will then be automatically imported and its class and slots names displayed in the **keywords** list.

3.2.2 Importing The Ontology from Another Project

To import the ontology from a separate project, in the **Original** tab, press the **Import from External KB** button. The system then asks for the Protégé Project file (extension ".pprj") from which to import the names. Use the displayed dialog box to locate the desired file, select it and press the **Open Protégé KB** button; or press **Cancel** to cancel the operation. If the ontology is imported successfully, its class and slot names are displayed in the **keywords** list.

3.3 Importing Rules

To complete the rule abstraction process, PJMappingTab requires the rules that are to be abstracted. These are imported to PJMappingTab in the following way:

1. In the **Original** tab, select the **Original Rules** tab in the lower half of the screen. This is where the original rules will be added. To do this either
   - Click in the text area to ensure the cursor is there (there will be a flashing vertical bar in the top left corner), and press **Ctrl+v** to paste the rules from the system clipboard.
     Note: this method requires the rules to have been copied to the system clipboard before performing this step. Alternatively,
   - Press the **Load Rules** button. This allows a file to be selected, whose contents will be read in, and assuming they are read successfully, the contents will be displayed in the **Original Rules** text area.

3.4 Creating Abstract Rules

The final step of the rule abstraction process requires the class and slot names and slo data type information from the ontology to be merged with the rules, so that the rules can be automatically marked to produce a set of abstract rules. To perform the rule abstraction, press the **Make Abstract Rules** button (located under the Keywords list).

It may be the case that the **Make Abstract Rules** button has not been automatically enabled. This is due to PJMappingTab not detecting that both the rules and the ontology have been imported. If this happens the simplest solution is typically to import the ontology again.

Assuming the abstraction process has been completed successfully, the new set of rules is displayed in a text area in the **Abstract Rules** tab. If this has been completed successfully each class and slot name in the rules will be slightly

modified, as will each variable name. Each class and slot name will have had a special prefix added to it to indicate that it is a class or slot name, and each slot name will have the slot data type appended to it. Similarly each variable now has a special prefix to indicate it as a variable name within the rules.

3.5 Saving Abstract Rules

For the new abstract rules to be of any use, they need to be saved to a file so they can be stored, distributed and used elsewhere against other ontologies. There are two ways of achieving this:

- Ensure the abstract rules are being displayed by clicking the **Abstract Rules** tab, press the **Save Rules** button. PJMappingTab then requests the name of a file to save the rules to, so navigate to where the file should be saved, enter a file name and press the **Save Rules** button. To cancel the save operation, press the **Cancel** button. Alternatively,
- Select the rules within the text area of the Abstract Rules tab (either with the mouse, or ensuring the focus is within the text area, with **Ctrl+a**) and press **Ctrl+c** to copy the rules to the system clipboard. They can then be copied from there into a text editor program and saved from there.

3.6 Performing Rule Mapping

The following section describes the second stage of the reuse procedure of PJMappingTab. This stage may be viewed as the reverse of the abstraction process, and involves taking a set of abstract rules, an ontology (from a Protégé project) and mapping the abstract rules to that ontology. Specifically the class names mentioned in the abstract rules are mapped to class names in the ontology, and slots mentioned in the abstract rules are mapped to slots in the ontology. PJMappingTab not only provides some assistance in the mapping procedure by attempting to automatically map between the classes and slots based on their respective names, but it also provides a graphical interface for performing the mapping manually.

3.7 Importing The Ontology to Map To

In order to map the abstract rules it is necessary to import the ontology on to which they are to be mapped. As with abstraction the ontology is imported from a Protégé project, achieved in the following way:

1. Select the **Mapping Keywords** tab. There are now two options: import the ontology from the project currently loaded in Protégé, or from a separate Protégé project.
   - To import the current ontology, press the **Import from Current KB** button. The ontology is then automatically imported.

- To import the ontology from a project saved on disk, press the **Import from External KB** button. PJMappingTab then asks for the Protégé Project file (extension ".pprj") from which to import the ontology. Use the displayed dialog box to navigate to the desired file, select it and press the **Open Protégé KB** button; or press **Cancel** to cancel the operation.

If the ontology is imported successfully, the names of its classes and slots are displayed in the **keywords** list.

3.8 Importing Abstract Rules

Before a set of abstract rules can be mapped to the ontology they must be imported into PJMappingTab. To accomplish this Select the **Abstract Rules** tab (located at the bottom half of the **Original** tab) Load the rules from a file, by pressing the **Load Rules** button, locate and select the desired file and press the **Load Rules** button. The contents of the select file are then read and displayed in the text area of the Abstract Rules tab.

Note: if performing mapping directly after completing the rule abstraction process on the newly abstracted rules this stage can be skipped.

3.9 Mapping

Once both the abstract rules and the keywords have been imported, the mapping interface is displayed in the **Mapping** tab, and can be used to aid in the mapping by having PJMappingTab search for and suggest possible matches based on lexical analysis and/or by manually defining mappings.

The mapping interface is split into two main sections: at the top are details of the individual mappings available, while the bottom displays the current state of the rules, which are updated every time the mappings are executed.

3.9.1 Automatic Mapping

Once the abstract rules and the mapping keywords have been imported, PJMappingTab searches the abstract rules, extracting the classes and slots defined within them, along with the slot data types, and suggests mappings between these extracted classes and the classes and slots imported from the ontology, displaying the results in the **Mapping** tab. This tab can then be used to complete and/or correct the suggested mappings and execute them upon the abstract rules.

The suggested mappings are displayed in the **Concept Mappings** tab of the **Mapping** tab. The details of the suggested mapping for each extracted class is displayed, which includes the suggested mapping for the class name, along with the suggestions for each of the classes slots. If PJMappingTab has been unable to determine a mapping for any slot or class, the target description is "has no suggested mapping," and will required to be mapped manually.

3.9.2 Setting the Automatic Mapping Properties

There are two properties that can be set with regard to the automatic mapping functionality of PJMappingTab. These are the location of the dictionaries for WordNet (used in looking up alternatives for the keywords) and the word constituent limit (see below for an explanation of the word constituent limit).

Setting the WordNet Dictionaries The WordNet dictionaries are crucial for the production of accurate suggestions by the automatic mapping function. Although it may work without them, in order to achieve best results they are required. Sometimes the dictionaries are not located on the hard disk where PJMappingTab expects them to find them (by default they should be located at ¡Protégé-home¿\plugins\ uk.ac.csd.abdn.pjmappingtab\wordnetdata\"), and so PJMappingTab needs to be updated with their current location. To update the location of the dictionaries

1. Press the **Auto Map Properties** button on the Mapping tab,
2. In the dialog that is displayed, the current location of the WordNet dictionaries is given in a text field labelled **JWordNet Dictionary location**. Press the **...** button to the right of the text field,
3. Use the file choice dialog to navigate to and select the folder containing the dictionary files,
4. Press **Open**, Check the location displayed is correct, if it is press **OK**, otherwise repeat steps 2 and 4.

Setting the Word Constituent Limit The word constituent limit refers to the percentage of constituents two strings must have in common before a mapping is proposed between them. The constituents of a string are the words that make up that string (for example the string *car_repair* has *car* and *repair* as its constituents). This level is set as a percentage of the total constituents in each string. To set this value

1. Press the **Auto Map Properties** button on the Mapping tab,
2. Use the **Word Constituent Limit** spinner to change the value to the desired level,
3. Press **OK**,
4. To cancel the operation at any time, press **Cancel**.

3.9.3 Manual Mapping

Depending on the original application of the rules and the target ontology, it may be the case that the automatic mapping facility suggests mappings for only a subsection of all the classes and slots used in the abstract rules. Additionally, it may be the case that the suggested mappings are incorrect and need to be

changed to achieve the desired functionality. Both of these tasks can be achieved by using the manual mapping facilities of PJMappingTab.

Class and Slot Mappings Each class and slot mapping has to be performed individually. Each mapping is divided into two components: the name of the class/slot as displayed in the rules that is to be mapped and a list of the relevant names in the ontology that it can be mapped to. To change a suggested mapping

1. For the class/slot that is to be changed, press the **Alt. Mappings for X** button where X is the name of the class (if it's a slot being changed, x is the name of the class the slot belongs to), located below the details of the mapping for the relevant class.
2. This displays the alternative mapping dialog, which displays a selection of alternative mappings which were created during the automatic mapping stage (do not worry if there are no other mappings displayed, it simply means no other mappings were determined). The **Name** column lists the class/slot name the suggestions on the same row match up to (the suggestions are in the **Suggestion 1**..**Suggestion n** columns. To use an alternative mapping, select the radio button on the bottom row of the column which contains the desired mapping.
3. If none of the alternative mappings are suitable, or no others are suggested, it is possible to define a mapping manually. To do this use the drop down lists in the **Manual** column.
   (a) First select the desired class from the top drop down list. The other drop down lists (for the slots) automatically update to contain only slots that are suitable for the slot that is being mapped. This is to ensure slots in the rules with a certain type are mapped to slots of the same or compatable slots.
   (b) Use the remaining drop down lists to select the target slot for each of the slots.
   (c) Ensure the radio button at the bottom of the Manual column is selected.
4. Once the desired mapping has been selected (the radio button below it will be a filled circle) press the **OK** button to confirm the change, or **Cancel** to cancel it.
5. The details of the mapping in the **Concept Mapping** tab will now be changed to reflect the new mapping.

Variable Mappings Although not required by JESS, it is sometimes desirable to change the names of the variables in the rules to make the rules clearer. This step is not necessary as the prefix's are set during the abstraction process to indicate that they are variables to JESS (as well as PJMappingTab), and are set to reflect if the slot is a multislot or not. The variable mappings are listed in the

Variable Mappings tab, and can be altered by in the following way

1. Select the **Variable Mappings** tab, located within the **Mapping** tab.
2. Enter the desired variable name in the text field directly to the right of the variable name that is to be changed. Take care to ensure the correct variable indicator is added ("\$\?" for a multislot and "\?" for a regular slot) at the start of the name. Do not worry about the slightly irregular pattern, as the '\' is removed during the mapping process.

3.10 Executing the Mappings

Once all the class, slot and variable mappings have been defined, the rules need to be updated to reflect them. During this process, all the abstract names in the rules are changed to the names of the class/slot they have been mapped to, as are the variable names. To execute the mappings press the **Execute Mappings** button in the Mapping tab. After a short pause, while the rules are updated, the new rules are displayed. The main cause of this pause is that, PJMappingTab invokes the automatic mapping algorithm again, to ensure consistency between the rules and the display.

3.10.1 Saving the New Rules

Once all the class and slot mappings, as well as any desired variable mappings, have been executed the rules can then be used via the JessTab against the ontology that they were mapped to. To achieve this, the rules must first be removed from PJMappingTab and entered at the JessTab console as normal. The rules can be removed by one of the following:

- Press the **Save Rules** button in the **Mapping** tab. Use the resulting dialog box to select a file in which to save them or enter a file name in the File Name field; and press the **Save Rules** button. Alternatively,
- Select the rules within the text area (either with the mouse, or by pressing **Ctrl+a** when the cursor is within the text area), and press **Ctrl+c** to copy the rules to the system clipboard, from where they can then be pasted into a text editor or directly into the JessTab console.